Smart Style on the Semantic Web*

Jacco van Ossenbruggen
CWI
P.O. Box 94079
1090 GB AMSTERDAM
The Netherlands
Jacco.van.Ossenbruggen@cwi.nl

Lynda Hardman
CWI
P.O. Box 94079
1090 GB AMSTERDAM
The Netherlands
Lynda.Hardman@cwi.nl

ABSTRACT

Web publishing systems have to take into account a plethora of Web-enabled devices, user preferences and abilities. Technologies generating these presentations will need to be explicitly aware of the context in which the information is being presented. Semantic Web technology can be a fundamental part of the solution to this problem by explicitly modeling the knowledge needed to adapt presentations to a specific delivery context. We propose the development of a *Smart Style* layer which is able to process metadata that describes content and use this metadata to improve the presentation of the content to human users. In the paper, we derive the requirements of such a Smart Style layer by considering Web design from both the document engineering and graphic design perspectives. In addition, design trade-offs made by human designers have to be taken into account for the automated process. After stating the requirements for a Smart Style layer, we discuss to what extent the currently available Web technology can be used and what its limitations are. The limitations are illustrated with examples of potential future extensions.

Keywords: Semantic Web, Device Independent Authoring, Document Engineering, Graphic Design.

Word count: 7150

1. INTRODUCTION

As the Web continues to grow not only in size but also in complexity, the increasingly varying needs of the intended audience marks the end of the "one size fits all" era. Delivery contexts [27] can be characterized in terms of specific user preferences and abilities, capabilities of the access device and available network resources. Given this heterogeneity, any single message needs to be adapted to a particular set of circumstances. As a minimum requirement, the author's intended message needs to be conveyed to the user given the constraints imposed by the access device. In addition, the generated presentation should conform as much as possible to the preferences of the user and the author [6]. These two types of adaptation may lead to an explosion of potential delivery contexts with which current stylesheet technology is unable to deal.

In previous work, we describe our prototype multimedia presentation generation system Cuypers [21]. Cuypers generates multimedia presentations adapted to the constraints of a specific delivery context. We claim that the particular solutions deployed within Cuypers realize a level of adaptivity that should become generally available on the Web. This introduces new challenges since the solutions need to be embedded within the current Web infrastructure. In this paper, we introduce the concept of *Smart Style*: an intelligent presentation adaptation layer for the Web that builds upon two fundamental technologies:

1. Web document engineering technology, including delivery formats such as HTML [30], SMIL [29], SVG [10] and XSL [28], and style and transformation languages such as CSS [4] and XSLT [7].
2. Semantic Web knowledge representation and metadata technology, including RDF [24], RDF Schema [25], DAML+OIL [19] and CC/PP [26].

Currently, Semantic Web technology is primarily deployed to improve Web-based information *gathering* and *brokerage*, with little attention to improving information *presentation*. Our vision is, however, that the Semantic Web infrastructure should not only play a key role in finding information on the Web, but also in presenting this information in the most appropriate way to each individual reader. Our proposed Smart Style layer will deploy Semantic Web technology to improve the presentation's adaptation, aiming for an optimized design of the presentation that suits the specific requirements of the user's delivery context.

In this paper, we derive the requirements for realizing the Smart Style layer. In section 2, we specify the key design ingredients of a Web-based presentation from two perspectives: a document engineering and a graphic design perspective. These allow internal trade-offs to be made in the design of a presentation. In addition, external forces that influence the decisions made during the design process are discussed in section 3. Both sections contribute to a set of requirements for Smart Style. Section 4 states to what extent the requirements are met in terms of the current Web infrastructure, identifies gaps and gives suggestions for extending the current Web infrastructure.

*This is a revised version of CWI Technical Report INS-R0201

Semantic Web Workshop 2002 Hawaii, USA

2. DESIGN PERSPECTIVES ON WEB PRESENTATIONS

In this section we compare two different perspectives of creating a presentation: document engineering and graphic design. The former assumes that the authoring process can be broken down into a sequence of sub-processes which are able to operate independently to generate the end result. The latter assumes a content-provider with a message to be communicated to a target audience, both of which the designer has to understand exactly before creating the appropriate mix of graphics and text to effectively communicate the client's message. Both perspectives are valid and need to be understood before distilling the requirements for a Smart Style layer.

2.1 Document Engineering Perspective

From a document engineering perspective, it is important to separate *content* from *style* information. The underlying principle is that the essence of the message is contained in the (XML-structured) text and remains unchanged when style parameters, such as screen width or font size, are varied. This principle allows the creation of an infrastructure where the file containing the content, the XML file, can be created and maintained separately from a style file, such as a CSS stylesheet. The advantages of this approach are well-known within the Web community. These include the reuse of the same content in different contexts and the enforcement of consistent styles across different sets of content [20].

A presentation, however, involves more than applying an appropriate style to the selected content. A third, and essential, ingredient is the *structure* of the presentation. The simple separation of content and style as described above suffices only when the presentation structure is similar to the content structure in the underlying XML. If this is not the case, then a transformation step, such as enabled by XSLT, is needed to convert the content structure to the desired presentation structure. For example, the lexical order in a source HTML document might need to be transformed to the order that is most appropriate in the text-flow of the target HTML presentation. Alternatively, a more structural process may be needed, such as a transformation of an XML document into an XSL formatting object tree.

The document engineering process of creating Web presentations can be summarized in three steps:

1. select or create the content (typically structured using XML);

2. define a mapping from content to the presentation structure that defines, among other things, the most appropriate order (e.g. by using XSLT);

3. (optionally) refine this presentation structure by applying preferred style parameters (e.g. by using CSS).

Essential in this approach is the assumption that the three steps can be carried out independently. Content can be entered into a database by a content-provider. This content can then be extracted from the database in the desired order by a server-side script written by a Web-site programmer. Finally, the preferred style parameters can be determined (server-side) by a graphic designer's and/or (client-side) by the end-user's stylesheet. For many (database) content-driven Web sites, this assumption holds. The same applies to knowledge-driven or model-driven sites (See for example, [12]). Furthermore, the current Web infrastructure, with its large number of XML-related tools, is well equipped to support this process.

2.2 Graphic Design Perspective

Despite the advantages of the document engineering approach, it also has significant limitations. Specifically, in our own work on automatically adapting multimedia presentations to a variety of delivery contexts, generic XML tools proved to be inadequate (see [21] for details). Current tools are unable to deal with multimedia content for which it is not known *a priori* which transformation and stylesheet are suitable for displaying the content in a particular context. In online multimedia databases, for example, multimedia presentations can be generated from the media items returned by a database query. Since information about the media items such as quantity, type, size and size is not known in advance, template-based solutions cannot be used for determining a suitable presentation structure. Several solutions for this problem have been proposed and include: the use of large numbers of templates, where selection of the correct template becomes a problem [9]; constraint-based approaches, using grammars [31], planners [1] or logic programming [21] to generate the constraints; and other model-driven approaches to automatic presentation generation [3, 13].

Presentation structure plays a much more important role in multimedia than in text-based applications. Multimedia users experience presentation structure primarily as the sequential arrangement of the constituent media items in time, as the spatial arrangement on the screen and as paths of navigational hyperlinks. The presentation structure of multimedia is more difficult to determine automatically by stylesheets. For text, stylesheets may change the layout (e.g. switch from single column to two column, or change the margins) while preserving the semantic integrity of the underlying message. For multimedia, changes in the spatio-temporal arrangement will often have a large impact on the perceived semantics of the presentation [22]. Multimedia formats such as SMIL [29] address this problem by allowing the author to specify the presentation structure explicitly in the document. This is required because in multimedia the message is conveyed not only by the individual media items but also by the spatio-temporal and navigational arrangements of the media items in the presentation. In multimedia the presentation structure and content are in general *not* independent.

The document engineering approach thus needs to be refined for media-centric applications, in which the assumption that content, presentation structure and style are independent is false. In contrast to the content-centric approach in most of the document engineering literature, most of the graphic design literature features a more balanced perspective on the relation between content, presentation structure and style and the roles these three ingredients all play in conveying the overall message of the presentation. Understanding these roles and their dependencies is crucial for determining the requirements of a Smart Style layer.

Figure 1 illustrates how decisions made in any of the three sections can influence the other two. We give examples of how each part of the figure influences the other two parts.

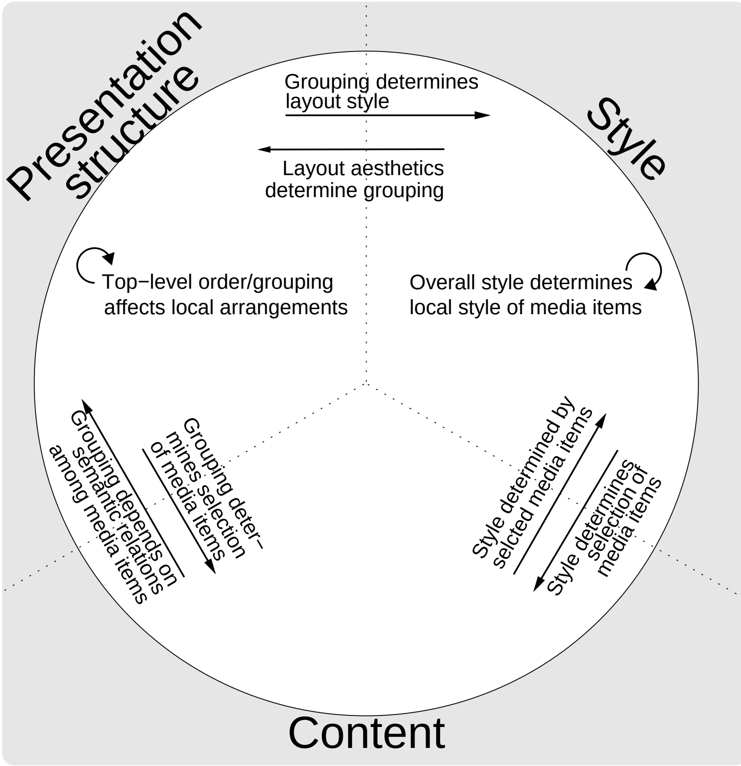

Figure 1: Dependencies between content, presentation structure, and style.

Presentation structure depends on content. The particular selection of content items can be used to determine the presentation structure of the items. For example, suppose that a number of media items have been selected for presentation. The items fall into three categories, and the presentation structure reflects these categories by first displaying all items in a single category before displaying the items from another category. For printed graphics, Williams [32] advocates the use of spatial layout to express grouping relationships in the underlying content. In multimedia time, space and links can all be used to communicate the underlying grouping relationships [18].

Style depends on content. The content can also influence the overall style of the presentation. For example, suppose a number of images are selected to convey the message. They happen to share a number of color characteristics, which lead to the choice of particular colors for the background and main text colors of the presentation. Other aspects, such as image texture, could, for example, also influence the selection of appropriate font type faces. Availability of appropriate content may influence the choice of style through, e.g., using an image as a background for the rest of the presentation.

Content depends on presentation structure. In the document engineering perspective, presentation structure is derived from the original content structure. In practice, however, when a Web site is created, the presentation structure, in particular the spatial layout and navigation structure among sections, is often determined first and the content created to fit into it. It is difficult, however, to make these dependencies explicit.

Style depends on presentation structure. The style can also depend on the presentation structure. For example, if the presentation structure uses spatial alignment for conveying grouping relationships in the content, then the designer needs to choose a particular alignment style (e.g. left-, centered or right-aligned). In figure 2, for example, the title is centered above the text to convey the grouping relation, in this case that the title applies both to the explanatory text and to the example image. Note that the same presentation structure could have been conveyed using a longer title aligned left with the *chiaroscuro* explanation.

Content depends on style. In the document engineering perspective, style is often perceived as the "add-on" after the "important" decisions have been made. The "more fundamental" choice of content may, however, also depend on the style. For example, in order to preserve the visual unity in a presentation, relevant images may be selected for presentation only if their color histograms or clarity of images fit in with the style of the presentation as a whole [17]. For Web-sites aiming for a strong visual effect (e.g. for branding purposes), the look-and-feel of the site is typically designed first. The content is selected, manipulated or created to fit in with the chosen style.

Presentation structure depends on style. Similarly, an established style may prescribe certain limits on the spatial grid and pacing of the presentation. Ideal groupings and orderings of selected content may have to be put aside for reasonable alternatives which fit in with consistent use of margin widths and item alignments throughout the presentation. Similar effects and tensions are possible for temporal layout. A rhythmic presentation gives a certain desired effect but may clash with specific durations needed to express the message at different points in the presentation.

. In addition to the mutual dependencies between each of the three aspects, local presentation structure or style can depend on more global presentation structure or style. This can be used to provide continuity and consistency throughout the presentation.

In summary, in the graphic design perspective of creating a presentation, aspects of content, presentation structure and style depend on each other in ways that are generally ignored in the document engineering perspective. This is not to say that document engineering tools are not useful, but rather that the extra dependencies which make the task of good design so complex require more complex solutions. Since automated adaptation requires finding solutions within this design space, the three aspects of content, presentation structure and style need to be expressed and manipulated explicitly.

Being aware of the internal mutual dependencies is necessary but insufficient for finding a satisfactory solution in the design space. The adaptation also needs to fulfill external requirements of, e.g., the user and content provider. In the following section we discuss potential external influences on the design process.

3. EXTERNAL FORCES ON THE DESIGN PROCESS

The previous section explains the mutual dependencies that play a role when making decisions about content, presentation structure and style. So far, we have limited the discussion to the dependencies that are *internal* to the process of designing a Web presentation. In this section we discuss the dependency of design decisions on *external* forces.

The external forces that influence the design originate directly from the different interests of the parties involved. To determine the requirements of an automated system, we use the following motivating example, based on a typical scenario with three main parties: a *content-provider* who wishes to effectively communicate a message to a *user*, aided by a skilled *designer*.

Examples of forces that originate from the content provider include the *mission* of the content provider's organization (e.g. making profit by selling books online), the limited availability of *resources* (e.g. the amount of time and money the organization is willing to spend on the design, the amount of disk space or bandwidth that is available at the server), and the content provider's *preferences* (e.g. the use of company colors in the Web forms).

Examples of forces that originate from the user include the user's *needs* (e.g. the desire to buy a book), the limitations imposed by the user's *delivery context* (e.g. the user could be driving a car, have a low bandwidth connection, have physical disabilities, or have strict time constraints), and the user's personal *preferences* (e.g. user could prefer visual to textual information, dislike fast cuts in video material, prefer soft colors to primary colors).

Given a good understanding of the type of forces that play a role, it is the task of the designer to come up with a design that best matches the needs of the content provider and the user. In addition to the forces originating from the content-provider and the user, there are additional forces originating from the designer, whose resources are also limited and might also have personal preferences. Many of these forces could give rise to conflicts and will require the designer to make balanced trade-offs. For example, the designer might decide *not* to use the soft colors of the organization's company logo for users that need to fill in Web-forms while working in bad lighting conditions.

The role of an intelligent automatic adaptation mechanism is very similar to that of the human designer. Automatic adaptation also has to deal with forces originating from content-provider and user, as well as with forces originating from the adaptation process itself (e.g. limited computing resources, or personal preferences of the developer of the adaptation system). For example, in figure 2 the preferred design centers the title across the width of the screen. In figure 3, however, the client's platform display is shorter and the images are not to be scaled — a condition imposed by the copyright holder to preserve image quality. This forced our Cuypers system to search for an acceptable layout design alternative within the given device and content-provider constraints. We do *not* claim that we can build an automated system that could make such decisions as well as a professional designer. Intelligent adaptation systems such as Cuypers can, however, make acceptable design decisions when dealing with these types of trade-offs. Their intelligence is based on explicit knowledge about the design space dependencies and external constraints, combined with an adequate search strategy. These characteristics require that adaptation be more than the application of a simple mapping from source to destination format. Rather, it requires heuristic reasoning to find an optimal solution to balance the forces involved.

The requirements for automated adaptation following from the discussions here and in section 2 can be stated as follows.

- Explicit knowledge reflecting the internal design dependencies among content, presentation structure and style discussed in section 2, and external influences of the delivery context specifying the user's resources, preferences and needs, and the content provider's server context.

- A transformation method which can take the above knowledge into account, to make an informed choice in the internal design space while balancing the external trade-offs. Note that this transformation method will not be based on simple mappings.

In the next section we discuss the implications of these requirements for extending the current Web infrastructure.

4. TOWARDS INTELLIGENT STYLESHEETS

The type of adaptation that can be found on the Web today may seem to be a far cry from the type of intelligent adaptation discussed in the previous sections. To a certain

Figure 2: Example SMIL presentation (generated by Cuypers for display on large screen).

extent, however, the current Web infrastructure already provides a good basis upon which a smarter adaptation layer could be built. Making this layer work in practice, however, requires the specification of new standards, and — arguably more difficult — a sophisticated and seamless embedding of these new techniques in the current Web framework.

In this section, we give a short overview of how current Web standards relate to automatic adaptation. We then discuss the requirements of a Smart Style layer illustrated by examples of potential extensions of current Web technology. We use the current Web Recommendations as a basis, and incorporate, as much as possible, other W3C activity that is still work in progress.

4.1 Current Web adaptation techniques

To a certain extent, a Web site can already build its own server-side adaptation techniques by deploying generic Web technology, such as CSS and XSLT stylesheets, which can be used to adapt and style XML and HTML content. These techniques can be combined with commonly used solutions such as filling Web templates with material stored in databases.

Today's Web formats also allow client-side adaptation. Many delivery formats, including HTML, feature basic functionality to improve accessibility. A well-known example is the `alt` tag that can be used to provide an alternative, textual description of the role of an image within an HTML page. SMIL features a more sophisticated example in the `switch` element, which can be used by multimedia authors to provide alternatives for parts of the presentation depending on the delivery context. The XSL vocabulary [28] includes features that allow similar client-side adaptation, including the `role` property and the `multi-switch` formatting object. The use of metadata also has a huge potential for improving automatic adaptation. A good example is the work on automatic linearization of SVG documents, to allow synthesized speech browsing of SVG [16, 11].

The appropriate use of the techniques sketched above already requires human designers to deal with the design dependencies discussed in section 2 and trade-offs discussed in section 3. For example, multimedia authors have to make trade-offs between function and resources for each `switch` element in their SMIL presentations, because they need to match the functionality of their media items against the available bandwidth and others resources that are required by these media. The trade-offs between function and preferences in the company color scheme example are similar to the decisions made when using CSS `!important` rules, where designers need to think to what extent the preferences of the end-user's stylesheet are allowed to override the defaults determined in the document's stylesheet.

These current techniques suffice for applications in which a human designer has, during authoring time, sufficient information for making the required design decisions and trade-offs. For applications for which this information is only available at the time of the user request, the decisions need to be made by the adaptation system. The current Web infrastructure is, however, insufficient for making run-time design decisions. In order to move towards the intelligent adaptation and styling advocated in this paper, we need to extend the current Web framework.

4.2 Communicating delivery contexts

The first ingredient we need is a commonly agreed upon way to communicate the information upon which we will base our design decisions. A key requirement is the ability to communicate delivery contexts. Delivery contexts are required in order to provide information about the client-side

Figure 3: Same SMIL presentation, but adapted to the smaller height of the user's display.

resources that are available and about the personal preferences of the user. Assuming that at least a part of the adaptation will need to take place on the server, it is *essential* to standardize the communication of delivery contexts: clients need to be able to send the information in a way that the server understands. A machine-readable description of a delivery context that can be sent to the server is often called a *profile*. Within W3C, work on a common ground for delivery contexts is currently in progress. CC/PP [26] provides an RDF-based framework for defining the vocabularies that are needed to define profiles. In addition, it also provides a small vocabulary that can be reused across different profiles. A typical example of a CC/PP profile is the User Agent Profile developed by the WAP Forum [33]. This profile provides a commonly agreed upon mechanism to communicate the (technical) capabilities of mobile phones to servers and proxies. The CC/PP framework, however, is sufficiently flexible to allow the definition of profiles that focus on more user-centered aspects of a delivery context.

From a technical point of view, CC/PP is built on top of RDF. CC/PP profiles use RDF statements to describe the relevant client-side capabilities and preferences. For example, figure 4 shows a fragment of a delivery context that uses CC/PP to inform the server that the client platform features a 640x480 display.

CC/PP profiles are, at the time of writing, hardly used on the Web (the WAP industry forms a notable exception). Communicating delivery contexts between client and server needs to become standard practice, which is more than an implementation issue. Additional CC/PP vocabularies need to be provided, not only to describe the capabilities of the hardware and software of the user's device, but also to describe the needs, environment and personal preferences of the user.

```
...
<ccpp:component>
  <rdf:Description rdf:about="TerminalHardware">
      <rdf:type rdf:resource="HardwarePlatform" />
      <ccpp:pix-x>640</ccpp:pix-x>
      <ccpp:pix-y>480</ccpp:pix-y>
      ...
  </rdf:Description>
...
</ccpp:component>
```

Figure 4: Example fragment of a delivery context specified using CC/PP.

4.3 Supporting metadata for content description

Clients need to be able to communicate delivery contexts, but in itself this is insufficient. Many design decisions will also depend on information that is available at the server-side. Even when this information is not intended to be published on the Web, having commonly used and standardized solutions for describing and processing it will greatly reduce the development effort needed to implement a smart, adaptive Web site.

Intelligent adaptation systems will need some knowledge of the function of the content they are adapting. To make this type of knowledge explicit, appropriate use of *metadata* will be of key importance. Within and outside W3C, a large amount of work on metadata standardization is currently in progress, and in most of this work RDF plays a central role. For example, work on RDF Schema aims at adding functionality that allows RDF vocabularies to be defined in a standardized way. Ontology languages, such as DAML+OIL, built on top of RDFS, add features while still allowing efficient implementations that are able to reason about metadata information.

While the current focus of this type of Semantic Web technology is on the use of metadata to achieve a more intelligent model for Web-based information retrieval (e.g. improving search engines), the use of metadata in our Cuypers system shows that there is also a huge potential in applying this type of technology for improving the adaptation and presentation process. Through the use of metadata to make the intended semantics and function of the content explicit, adaptation systems are able to make informed decisions during the design process. For example, suppose an online museum site has developed an RDF Schema[1] for the metadata[2] used to annotate their Web site. Also suppose the site features an HTML page describing a work by the painter Rembrandt van Rijn, focusing on the use of *chiaroscuro* (the painting technique that uses strong contrasts of light and dark paintings). Figure 5 shows a fragment of the HTML version of the earlier SMIL presentation.

```
<div id="allegory">
 <h1>Musical Allegory</h1>
 <img src="allegory.jpg"/ >
 <p>This is hardly just an ordinary group of musicians.
    The figures are too exotically dressed in oriental
    ...
</div>
```

Figure 5: Example XHTML 1.0 fragment from a page about a Rembrandt painting.

From an XML markup perspective, all we know is that we have a fragment with a first level heading, an image and a text paragraph. The underlying semantics, however, could be explicitly added by the use of RDF metadata, as shown in figure 6.

```
<museum:Painter rdf:ID="Rembrandt">
  <museum:fname>Rembrandt</museum:fname>
  <museum:lname>Harmenszoon van Rijn</museum:lname>
  <museum:painted rdf:resource="#allegory" />
</museum:Painter>

<museum:Painting rdf:about="#allegory">
   <museum:title>Musical Allegory</museum:title>
   <museum:technique>Chiaroscuro</museum:technique>
</museum:Painting>
```

Figure 6: RDF metadata of XHTML 1.0 fragment.

This explicitly states that our HTML fragment is an instance of a class `Painting`, with a `title` property "Musical Allegory", and that there is a `Painter` instance that has a `painted` relation with the painting at hand.

Given such semantic information about the content, and the explicit descriptions of the delivery context, adaptation engines should be able to make better decisions about how to adapt the presentation to a particular situation. For example, because the metadata explicitly states that the painting is using the technique "Chiaroscuro", an adaptation engine might decide to add, for non-expert users, a link to the page describing this technique. This requires an adaptation process that takes into account both the delivery context (because it needs to know that the user is a non-expert) and metadata (because it needs to know in which conditions it should add a link). Based on our experience with Cuypers, we found that most metadata is used for content descriptions that are defined in terms of the application domain. This may be sufficient for most information retrieval purposes, but not for information presentation. Metadata that, for example, identifies the potential role the content could play in the presentation is hard to find. In the example above it was hard to predict on the basis of the metadata whether the textual description of the "Chiaroscuro" technique is suitable for non-expert users or not. For the images, it was hard to determine to what extent images could be resized to fit the presentation without compromising the information that was intended to be conveyed.

In general, to improve intelligent adaptation and presentation, metadata annotations of Web content is required. Annotation should, however, not be confined to information retrieval, but also facilitate information presentation.

4.4 Processing delivery contexts and content descriptions

Assuming that the information upon which we base our design decisions will be available from the Web through the use of standard Semantic Web technologies such as CC/PP and RDF, the next ingredient needed for building a Smart Style layer is an efficient set of tools that allows this information to be taken into account during the adaptation process. As described above, many of the current generation W3C Recommendations already have some features that address adaptation issues. A first step is to make the current generation presentation-oriented Web technology interoperable with the next-generation Semantic Web technology. For example, CSS stylesheets are currently not able to take CC/PP profiles into account. CSS has, however, a feature that is closely related to CC/PP, and allows the specification of device dependent style rules: the `@media` rule. Figure 7 shows an example[3] of a stylesheet that uses bigger fonts on computer screens than on paper printouts of the same document.

```
@media print {
  body { font-size: 10pt }
}
@media screen {
  body { font-size: 12pt }
}
```

Figure 7: Device dependent style rules as already supported in CSS2.

A first step towards a CSS syntax that allows more detailed queries is suggested in [15]. In this syntax, queries to specific device features are allowed. For example, the CSS media rule for screen display above could be further refined by adding constraints on the minimum width of the screen,

[1]Museum schema example adapted from [14].
[2]Metadata example adapted from [23]).
[3]Example taken from the CSS2 Specification [4].

as shown in figure 8. Using the constraints, stylesheets could take into account the information provided by profiles such as the example in figure 4.

```
@media screen and (min-width: 640px) {
  body { font-size: 14pt }
}
@media screen and (min-width: 800px) {
  body { font-size: 16pt }
}
```

Figure 8: Detailed media queries using a CSS3 extension (work in progress).

Even from this extended CSS syntax, however, it is still a long way to fully CC/PP aware style engines. CC/PP features that will affect style application include the ability to define new profile vocabularies, inheritance mechanisms for specifying default values and the description of the capabilities of transcoding proxies. Style engines need to be able to deal with these features in order to take full advantage of the information specified in CC/PP delivery contexts.

Note that the need to take CC/PP information into account also applies to XSLT transformation engines. While the full details of how this could affect future versions of XSLT are beyond the scope of this paper, one could, for example, imagine an extension[4] of XSLT's *mode* concept. For example, transformation rules could be selected in a way similar to that of the media rules in CSS. In such a hypothetical extension (see figure 9) one could, for instance, define a rule for creating a two column layout only if the output medium is print and the paper is wider than 17cm.

```
<xsl:template match="body"
              mode="print and (min-width: 17cm)">
  ...
  <fo:region-body column-count="2"/>
  ...
</xsl:template>
```

Figure 9: Device dependent rules by extending XSLT modes (tentative syntax).

In addition to taking information about delivery contexts into account, stylesheets also need to take into account the semantic information that is contained in the metadata associated with the content. Currently, style selector mechanisms only match on the *syntactic* properties of the underlying (XML) document hierarchy. This applies both to the selector mechanism used by CSS and to the XPath [8] selectors used by XSLT.

In all examples above, the rules were intended to match on the `<body>` element of an HTML document. Similar rules could be written to match on the syntactic properties of metadata, e.g. on the XML element and attribute names that are used to encode the RDF statements in figure 6. Using the current generation CSS and XSLT engines to process general metadata it is, however, not practical to match on the *semantic* properties of metadata: for CSS and XSLT processors, RDF is just XML. As a result, it is very hard to write, for example, a rule that matches on all alternative XML serializations that are allowed for RDF. A more serious problem, however, is that it is impossible to write CSS or XSLT rules that make use of the structural relations of RDF and RDF Schema, for instance a style rule that applies to all objects that are instances of a specific RDFS (sub)class. Neither is it possible to write rules for all objects that have a certain DAML+OIL-defined ontological relation, etc. Model-driven Web site management systems such as OntoWebber [12] are thus forced to develop their own solutions to associate presentation design elements to their RDF (and DAML+OIL) data, because CSS and XSLT are currently not applicable to RDF.

[4]We are not advocating a specific syntax, but are only claiming that future XSLT transformations need to be able to take CC/PP-like information into account

Future, Semantic Web-aware, selector mechanisms could allow specification of style rules in terms of the RDF semantics expressed in the metadata. This would extend the currently used CSS and XPath selectors, that are based on the XML syntax encoding the semantics. Consider the extended XSLT example rule in figure 10, which uses the RDF-aware query language RQL [14] for its selector, instead of XPath.

```
<xsl:template match=
  "RQL(http://www.museum.com/schema.rdf#Artifact)">
  ...
</xsl:template>
```

Figure 10: Semantic matching of XSLT rules using RQL selectors (tentative syntax).

The RQL query used would match on all instances of `Artifact` and its subclasses. Since our Museum RDF Schema defines `Painting` as a subclass of `Artifact`, the rule above would match on the semantics and structure of the RDF metadata describing the painting shown in figure 6, irrespective of the XML serialization syntax used to encode these semantics [5].

4.5 Beyond CSS and XSLT style and transformation rules

Above, we suggested extensions to CSS and XSLT that would allow stylesheets to take into account delivery contexts as specified by CC/PP and content semantics as expressed by RDF metadata. While taking this type information into account is a prerequisite for a Smart Style layer, this is in itself not sufficient.

Adaptation engines need to be able to search in the design space sketched in section 2, and make the trade-offs discussed in section 3. This type of decision process is hard to define using the simple "if *selector matches* then *apply rule body*" type of current style and transformation rules.

In addition, our experiments with the Cuypers system [21] allowed us to analyze the adaptation process of multimedia presentations for which the quantity, type and size of the media items were not known until run-time. We found that for these applications, automatic adaptation also requires the ability to verify the presentations that result from applying a set of transformation rules. When designing transformation rules for dynamic multimedia, one cannot, at authoring time, guarantee that the resulting presentation indeed meets the "hard" constraints imposed by the available resources. We have used the Cuypers system to experiment with a transformation engine that can evaluate the multimedia presentations it generates. The system employs backtracking to search for alternative rules when the end result does not meet the constraints imposed by the available resources. For example, even when a specific rule is applied only for target screens with a certain width, that condition in itself will not guarantee that the presentation resulting from applying the rule to media content of unknown size will indeed meet the maximum width constraints. What is needed is a means of evaluating the actual width of the final presentation, and a means of trying alternative rules in case the presentation did not meet the constraints.

While CSS and XSLT rules cannot be used to specify the required search strategies, this type of processing is vital for intelligent adaptive behavior on the Web. The Web thus requires more sophisticated ways of transforming the combined information provided by delivery contexts, metadata and the content into meaningful presentations. In future research, we want to explore how, and to what extent the combined search, transformation and evaluation techniques used within our Cuypers system could be made generally available on the Web.

4.6 Beyond atomic style properties

In addition to improved transformation processes, we also need to develop better abstractions to reason about the "soft" constraints imposed by the preferences of the parties involved. This type of reasoning requires explicit knowledge of the dependencies discussed in section 2. Taking these preferences and the associated dependencies into account will have a large impact on the perceived overall quality and design of automatic Web presentations. Currently, style rules work only on the basis of individual style properties. For example, one can specify the font type or color of a specific XML element. To what extent the application of these individual rules yield the desired overall result is hard to predict in advance, especially when dealing with more complex publishing systems that feature dynamic content, XSLT transformations, transcoding proxies and CSS stylesheets. After this process, the font and color of two XML elements positioned together in the final presentation might not go well together. Within the graphic design profession, style guidelines and checklists have been developed that can be used to avoid such design mistakes (see, for example, [32, 17]). It should be possible to build on this body of knowledge, and at least check the overall presentation against the most common design flaws. In addition to graphic design, similar checks could be developed for checking the design of the overall temporal flow of, and synchronization within, the presentation [2], and for checking the design of the navigation and interaction schemes that the presentation exposes to the end user.

5. CONCLUSIONS

Current Web technology addresses the problem of multiple delivery contexts through the use of CSS and XSLT stylesheets. These can be used for transforming presentation-independent XML documents to specific presentation formats, such as XHTML, SVG or SMIL. This is sufficient for dealing with a limited number of delivery contexts per stylesheet but is inadequate for adapting content to the plethora of delivery contexts for different devices, network resources and user groups. To solve this problem the Web is currently missing three key ingredients.

1. **Common vocabularies for describing delivery contexts** Web applications need to be able to communicate their capabilities and the preferences of their users so that transformation engines are able to make informed choices during the presentation generation or transcoding process. CC/PP already provides a framework for defining such vocabularies. Commonly agreed upon vocabularies will be needed for defining user preferences, device capabilities, network characteristics etc.

2. **Intelligent transformation methods** Transformations need to be able to take into account a wide variety of delivery contexts to generate a presentation corresponding to a particular delivery context. While it is unrealistic to expect that even an intelligent stylesheet would be sufficiently powerful to cater for *any* given delivery context, our claim is that the current transformation technologies can be significantly improved in order to allow a substantial increase in flexibility.

3. **Explicit metadata and design knowledge** Given the vocabularies for describing delivery contexts, and given an appropriate transformation method, in theory we would be able to develop adequate intelligent stylesheets. In practice, however, these stylesheets would implicitly contain a large amount of design knowledge and domain knowledge. This type of knowledge should preferably be made explicit and specified declaratively, in a similar manner to the explicit and declarative delivery contexts. RDF Schema [25] and DAML+OIL [19] already provide a framework for encoding this type of knowledge. To what extent vocabularies for this type of knowledge can be standardized remains to be seen, since they may be highly domain and application specific.

As these three ingredients build directly upon Semantic Web technology, we believe that only by a synthesis of (future) Semantic Web tools with the presentation-oriented tools of the (current) Web, can we hope to address the adaptation problems discussed.

This brings us to the first Achilles' heel of our Smart Style layer: the large amount of current and future W3C Recommendations that currently exist. Many of the Recommendations can be used to address part of the problem, but it is not clear how they can be used in concert to solve the overall problem. This paper derives the requirements for an ambitious goal: automatic adaptation of dynamic text and multimedia content to the requirements of the individual user's delivery context, while respecting the integrity of the semantics of the content. If we reduce our ambition levels, however, and "only" aim for taking into account processing context information, this alone would still have major consequences. To prevent CC/PP from becoming a standalone W3C recommendation that can only be processed with proprietary tools, we need to clearly define how other recommendations, including CSS, XSLT, XHTML, SMIL and SVG operate in the context of CC/PP. From CC/PP-aware

Web transformations, another step is required towards Semantic Web-aware transformations that also take metadata semantics into account. This will require tools that can abstract from the underlying XML syntax and operate directly on the semantics of languages such as RDF, RDFS and DAML+OIL. Realizing this level of interoperability among W3C Recommendations will be a huge effort. It should be clear that the examples given in this paper serve only to illustrate the derived requirements, and should by no means be regarded as readily applicable solutions to achieve the required interoperability. Making the current Web infrastructure interoperate seamlessly with the upcoming Semantic Web will be a huge challenge and a long term effort.

Finally, the other Achilles' heel of our Smart Style layer is the large amount of high quality design and domain knowledge that it requires. Smart Style does not aim at replacing human designers, but strives for providing applications with sufficient design knowledge when design decisions cannot be make by humans. It will require a large amount of human effort to make this knowledge explicit and it will require even more work to maintain it and keep it up to date. Given the problems most authors already have when they are forced to move from the "what you see is what you get" paradigm of desktop publishing to the "structured document" paradigm of XML-based Web publishing, this will not be an easy job. Having said this, we should also realize that we do not have to build it overnight: just as the current Web, we can build the Semantic Web with its Smart Style layer incrementally, by building new layers on the XML and RDF-based framework that is ready to be used now. Content-providers will start to use these new layers as soon as there are sufficiently large economic (e.g. attracting more customers by making their site accessible from new mobile devices) or legal (e.g. laws that require sites — including multimedia content — to be accessible for users with disabilities) incentives.

Acknowledgments

Many of the issues discussed were originally inspired by the W3C workshop on Device Independent Authoring in Bristol, October 2000. The research has been funded by the European ITEA/RTIPA and the Dutch Dynamo and ToKeN2000 projects. Examples are taken from a ToKeN2000 demonstrator, and all media content has been kindly provided by the Rijksmuseum in Amsterdam. Frank Cornelissen, Patrick Schmitz, Jeen Broekstra and Frank van Harmelen, and our CWI colleagues Joost Geurts, Oscar Rosell and Lloyd Rutledge provided many valuable insights.

6. REFERENCES

[1] Elisabeth André, Jochen Müller, and Thomas Rist. *WIP/PPP: Knowledge-Based Methods for Fully Automated Multimedia Authoring*. London, UK, 1996.

[2] Frederic Bes, Muriel Jourdan, and Farid Khantache. A Generic Architecture for Automated Construction of Multimedia Presentations. In *Proceedings of the International Conference on Multimedia Modeling 2001 (MMM01)*, pages 229–246, CWI, Amsterdam, The Netherlands, November 5-7, 2001.

[3] M. Bordegoni, G. Faconti, M.T. Maybury, T. Rist, S. Ruggieri, P. Trahanias, and M. Wilson. A Standard Reference Model for Intelligent Multimedia Presentation Systems. *Computer Standards & Interfaces*, 18(6-7):477–496, December 1997.

[4] Bert Bos, Håkon Wium Lie, Chris Lilley, and Ian Jacobs. Cascading Style Sheets, level 2 CSS2 Specification. W3C Recommendations are available at http://www.w3.org/TR, May 12, 1998.

[5] J. Broekstra, A. Kampman, and F. van Harmelen. Sesame: An Architecture for Storing and Querying RDF Data and Schema Information. In D. Fensel, J. Hendler, H. Lieberman, and W. Wahlster, editors, *Semantics for the WWW*. MIT Press, 2001.

[6] D.C.A. Bulterman, L. Rutledge, L. Hardman, and J. van Ossenbruggen. Supporting Adaptive and Adaptable Hypermedia Presentation Semantics. In *The 8th IFIP 2.6 Working Conference on Database Semantics (DS-8): Semantic Issues in Multimedia Systems*, Rotorua, New Zealand, 5-8 January 1999, 1999.

[7] James Clark. XSL Transformations (XSLT) Version 1.0. W3C Recommendations are available at http://www.w3.org/TR/, 16 November 1999.

[8] James Clark and Steve DeRose. XML Path Language (XPath) Version 1.0. W3C Recommendations are available at http://www.w3.org/TR/, 16 November 1999.

[9] Isabel F. Cruz, Kimberly M. James, and David C. Brown. Integrating Layout into Multimedia Data Retrieval. In *AAAI Fall Symposium on Using Layout for the Generation, Understanding, or Retrieval of Documents*, Cape Cod, Massachusetts, 1999.

[10] Jon Ferraiolo. Scalable Vector Graphics (SVG) 1.0 Specification. W3C Recommendations are available at http://www.w3.org/TR/, 4 September 2001.

[11] Ivan Herman. Follow up on SVG Linearization and Accessibility. Work in progress. Document available at http://www.w3.org/2001/svgRdf/, 19 July 2001.

[12] Yuhui Jin, Stefan Decker, and Gio Wiederhold. OntoWebber: Model-Driven Ontology-Based Web Site Management. In *Semantic Web Working Symposium (SWWS)*, Stanford University, California, USA, July 30 – August 1, 2001.

[13] Th. Kamps. *Diagram Design : A Constructive Theory*. Springer Verlag, 1999.

[14] Gregory Karvounarakis, Vassilis Christophides, Dimitris Plexousakis, and Sofia Alexaki. Querying Community Web Portals. http://www.ics.forth.gr/proj/isst/RDF/RQL/rql.html.

[15] Håkon Wium Lie and Tantek Ĉelik. Media queries. Work in progress. W3C Working Drafts are available at http://www.w3.org/TR, 17 March 2001.

[16] Guillaume Lovet. SVG Linearizer tools. Internship report and software download. Work in progress. Document available at http://www.w3.org/WAI/ER/ASVG/, 25 August 2000.

[17] Mark Oldach. *Creativity for Graphic Designers*. Narth Light Books, Cincinnati, Ohio, 1995.

[18] Lloyd Rutledge, Jacco van Ossenbruggen, Lynda Hardman, and Dick C.A. Bulterman. Structural Distinctions Between Hypermedia Storage and Presentation. In *Proceedings of ACM Multimedia*, pages 145–150. ACM Press, November 1998.

[19] Frank van Harmelen, Peter F. Patel-Schneider, and

Ian Horrocks. Reference description of the DAML+OIL (March 2001) ontology markup language. http://www.daml.org/2001/03/reference.html. Contributors: Tim Berners-Lee, Dan Brickley, Dan Connolly, Mike Dean, Stefan Decker, Pat Hayes, Jeff Heflin, Jim Hendler, Ora Lassila, Deb McGuinness, Lynn Andrea Stein, ...

[20] Jacco van Ossenbruggen. *Processing Structured Hypermedia — A Matter of Style*. PhD thesis, Vrije Universiteit, Amsterdam, The Netherlands, April 10, 2001. Also available on http://www.cwi.nl/~jrvosse/thesis/.

[21] Jacco van Ossenbruggen, Joost Geurts, Frank Cornelissen, Lloyd Rutledge, and Lynda Hardman. Towards Second and Third Generation Web-Based Multimedia. In *The Tenth International World Wide Web Conference*, pages 479–488, Hong Kong, May 1-5, 2001. IW3C2.

[22] Jacco van Ossenbruggen, Lynda Hardman, and Lloyd Rutledge. Integrating Multimedia Characteristics in Web-based Document Languages. Technical Report INS-R0024, CWI, December 2000.

[23] Jacco van Ossenbruggen, Lynda Hardman, and Lloyd Rutledge. Hypermedia and the Semantic web: A research agenda. Technical Report INS-R0105, CWI, 2001.

[24] W3C. Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendations are available at http://www.w3.org/TR, February, 22, 1999. Editied by Ora Lassila and Ralph R. Swick.

[25] W3C. Resource Description Framework (RDF) Schema Specification 1.0. W3C Candidate Recommendations are available at http://www.w3.org/TR, 27 March 2000. Edited by Dan Brickley and R.V. Guha.

[26] W3C. Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies. Work in progress. W3C Working Drafts are available at http://www.w3.org/TR, 15 March 2001. Edited by Graham Klyne, Franklin Reynolds, Chris Woodrow and Hidetaka Ohto.

[27] W3C. Device Independence Principles. Work in progress. W3C Working Drafts are available at http://www.w3.org/TR, 18 September 2001. Edited by Roger Gimson, co-edited by Shlomit Ritz Finkelstein, Stéphane Maes and Lalitha Suryanarayana.

[28] W3C. Extensible Stylesheet Language (XSL) Version 1.0. W3C Recommendations are available at http://www.w3.org/TR/, 15 October 2001, 2001.

[29] W3C. Synchronized Multimedia Integration Language (SMIL 2.0) Specification. W3C Recommendations are available at http://www.w3.org/TR/, August 7, 2001. Edited by Aaron Cohen.

[30] W3C. XHTML 1.1 - Module-based XHTML. W3C Recommendations are available at http://www.w3.org/TR/, May 31, 2001. Edited by Murray Altheim and Shane McCarron.

[31] L. Weitzman and Kent Wittenburg. Automatic presentation of multimedia documents using relational grammars. In *Proceedings of the second ACM international conference on Multimedia '94*, pages 443–451, San Francisco, October 15 - 20, 1994.

[32] Robin Williams. *The Non-Designer's Design Book*. Peachpit Press, 1994.

[33] Wireless Application Group. WAP-174: WAG UAPROF User Agent Profile Specification, 1999.